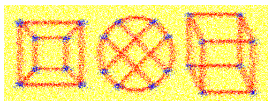

Journal of Graph Algorithms and Applications
http://jgaa.info/ vol. 22, no. 4, pp. 607–616 (2018)
DOI: 10.7155/jgaa.00480

A note on block-and-bridge preserving maximum common subgraph algorithms for outerplanar graphs

Nils M. Kriege Andre Droschinsky Petra Mutzel

Department of Computer Science
TU Dortmund University, Germany

Abstract

Schietgat, Ramon and Bruynooghe [18] proposed a polynomial-time algorithm for computing a maximum common subgraph under the block-and-bridge preserving subgraph isomorphism (BBP-MCS) for outerplanar graphs. We show that the article contains the following errors:

(i) The running time of the presented approach is claimed to be $\mathcal{O}(n^{2.5})$ for two graphs of order n. We show that the algorithm of the authors allows no better bound than $\mathcal{O}(n^4)$ when using state-of-the-art general purpose methods to solve the matching instances arising as subproblems. This is even true for the special case, where both input graphs are trees.

(ii) The article suggests that the dissimilarity measure derived from BBP-MCS is a metric. We show that the triangle inequality is not always satisfied and, hence, it is not a metric. Therefore, the dissimilarity measure should not be used in combination with techniques that rely on or exploit the triangle inequality in any way.

Where possible, we give hints on techniques that are suitable to improve the algorithm.

Submitted:	Reviewed:	Revised:	Accepted:	Final:
May 2018	August 2018	October 2018	December 2018	December 2018
		Published: December 2018		
	Article type: Concise Paper		Communicated by: G. Liotta	

This work was supported by the German Research Foundation (DFG), priority programme "Algorithms for Big Data" (SPP 1736), project "Graph-Based Methods for Rational Drug Design".

E-mail addresses: nils.kriege@tu-dortmund.de (Nils M. Kriege) andre.droschinsky@tu-dortmund.de (Andre Droschinsky) petra.mutzel@tu-dortmund.de (Petra Mutzel)

1 Introduction

Graph comparison is getting increasingly important with the growth of data analysis tasks on graphs and networks. An important application occurs in molecular chemistry for the tasks of virtual screening of molecular data bases, substructure search of molecules, and the discovery of structure-activity relationships within rational drug design. Thereby, finding the largest substructure that two molecules have in common is a fundamental task [12]. Since molecules can naturally be represented by graphs, the problem is phrased as maximum common subgraph problem, which is as follows. Given two graphs, find a graph with a largest possible number of edges that is isomorphic to subgraphs of both input graphs. This classical graph theoretical problem generalizes the subgraph isomorphism problem and is well-known to be NP-hard in general graphs [7]. Even deciding whether a forest G is isomorphic to a subgraph of a tree is an NP-complete problem [7]. However, if G is a tree the subgraph isomorphism problem can be solved in polynomial time [16, 17, 4, 21, 19]. The generalisation of this approach to the maximum common subgraph problem is attributed to J. Edmonds [16]. However, the vast amount of molecular graphs of interest are not trees, but outerplanar graphs, i.e., they admit a drawing on the plane without edge crossings such that all vertices are incident to the outer face of the drawing. Even deciding whether a tree is isomorphic to a subgraph of an outerplanar graph is NP-complete [20]. On the other hand, subgraph isomorphism can be solved in polynomial time when both graphs are biconnected and outerplanar [13]. More general, subgraph isomorphism can be solved in polynomial time in k-connected partial k-tree [15, 8].

Based on these theoretical findings, Horváth, Ramon and Wrobel [10] proposed to consider so-called *block-and-bridge-preserving* (BBP) subgraph isomorphism for mining frequent subgraphs in databases of outerplanar molecular graphs. The BBP subgraph isomorphism allows to consider blocks, i.e., the biconnected components, and the trees formed by the bridges separately and thereby can be solved in polynomial-time. Moreover, the approach yields chemical meaningful results, since it requires that the ring systems of molecules are preserved.

The maximum common subgraph problem in outerplanar graphs of bounded degree can be solved in polynomial time [1]. Although molecular graph have bounded degree and are often outerplanar, the algorithm has a high running time and is probably not suitable for practical use. Schietgat, Ramon and Bruynooghe [18] proposed to determine a maximum common subgraph under the BBP subgraph isomorphism and developed an algorithm with a claimed running time of $\mathcal{O}(n^{2.5})$ for two outerplanar graphs of order n. While the authors presented promising experimental results on graphs representing molecules, we show that their theoretical analysis of their approach is flawed. Moreover, we show that the proposed approach to derive a distance from the size (or weight) of the maximum common subgraph does not yield a proper metric.

2 Preliminaries

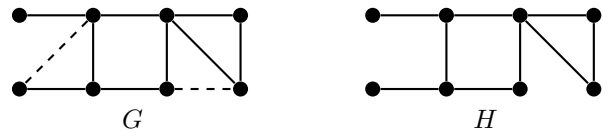
We briefly summarize the necessary terminology and notation. A *graph* $G = (V, E)$ consists of a finite set $V(G) = V$ of *vertices* and a finite set $E(G) = E$ of *edges*, where each edge connects two distinct vertices. A *path* of length n is a sequence of vertices $(v_0, \ldots, v_n)$ such that $\{v_i, v_{i+1}\} \in E$ for $0 \leq i < n$. A *cycle* is a path of length at least 3 with no repeated vertices except $v_0 = v_n$. A graph is *connected* if there is a path between any two vertices. A graph is *biconnected* if for any two vertices there is a cycle containing them. A *tree* is a connected graph containing no cycles. A graph G with an explicit root vertex $r \in V(G)$ is called *rooted* graph, denoted by G^r. A graph $G' = (V', E')$ is a *subgraph* of a graph $G = (V, E)$, written $G' \subseteq G$, if $V' \subseteq V$ and $E' \subseteq E$. A *block* is a maximal subgraph that is biconnected. An edge is a *bridge* if it is not contained in any block. A *matching* in a graph G is a subset of edges $M \subseteq E(G)$ such that no two edges in M share a common vertex, i.e., $e \cap e' = \emptyset$ for all distinct edges $e, e' \in M$. Given a bipartite graph G with edge weights $w : E(G) \to \mathbb{R}$, the *weighted maximal matching problem* asks for a matching $M \subseteq E$ in G such that the weight $w(M) = \sum_{e \in M} w(e)$ is maximal.[1]

An *isomorphism* between two graphs G and H is a bijection $\varphi : V(G) \to V(H)$ such that $\{u, v\} \in E(G) \Leftrightarrow \{\varphi(u), \varphi(v)\} \in E(H)$ for all $u, v \in V(G)$. We say that the edge $\{u, v\}$ is mapped to the edge $\{\varphi(u), \varphi(v)\}$ by φ. A *subgraph isomorphism* from a graph G to a graph H is an isomorphism between G and a subgraph $H' \subseteq H$. A graph G is said to be *subgraph isomorphic* to a graph H, written $G \preceq H$, if a subgraph isomorphism from G to H exists. A subgraph isomorphism from G to H is *block and bridge preserving* (BBP) if (i) each bridge in G is mapped to a bridge in H, and (ii) any two edges in different blocks in G are mapped to different blocks in H. We write $G \sqsubseteq H$ if a BBP subgraph isomorphism from G to H exists. A (BBP) *common subgraph* of two graphs G and H is a connected graph I such that $I \preceq G$ and $I \preceq H$ ($I \sqsubseteq G$ and $I \sqsubseteq H$). A (BBP) common subgraph I is *maximum* w.r.t. a weight function w if there is no (BBP) common subgraph I' with $w(I') > w(I)$. The two different concepts, maximum common subgraph (MCS) and BBP-MCS, are illustrated in Figure 1. The above definitions can be naturally extended to graphs with vertex and edge labels, where an isomorphism must preserve labels and the weight function may depend on the labels.

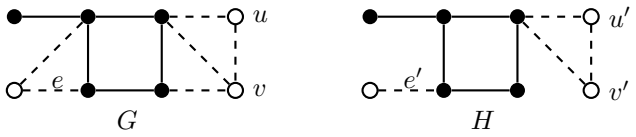
3 Complexity Analysis

The BBP-MCS algorithm for outerplanar graphs proposed in [18] decomposes the two input graphs into subgraphs with distinct root vertices referred to as *parts* (see Section 3.2 for a formal definition). An MCS problem for all compat-

[1] Note that in [18] matchings are defined as specific relations between sets, cf. Definiton 15. The running time to compute a matching then depends on the number of pairs with strictly positive weight. This can be expressed in a natural way by the number of edges in bipartite graphs.



(a) MCS I with $w(I) = 17$



(b) BBP-MCS I with $w(I) = 10$

Figure 1: Two graphs G, H and their MCS (a) and BBP-MCS (b), where $w(I) = |V(I)| + |E(I)|$. Dashed edges and blank vertices are not part of the common subgraph. Note that in Figure (b) the vertex at the bottom left cannot be included since e is part of a block in G and e' is a bridge in H. The two vertices u and v of G cannot be added, since the triangle containing u' and v' constitutes a distinct block of H.

ible pairs of parts is then solved using a dynamic programming strategy. Here, a series of weighted maximal matching instances arises as subproblems. It has been claimed [18, Theorem 2] that for two outerplanar graphs G and H the proposed BBP-MCS algorithm runs in time

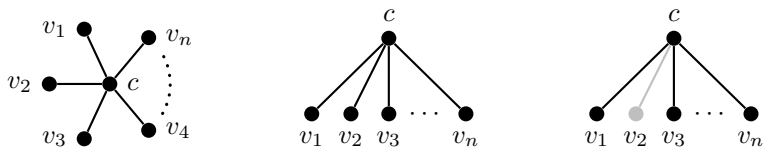
$$\mathcal{O}\left(|V(G)| \cdot |V(H)| \cdot (|V(G)| + |V(H)|)^{\frac{1}{2}}\right),$$

which is $\mathcal{O}(n^{2.5})$ for $|V(G)| = |V(H)| = n$. We show that this bound cannot be obtained by the presented techniques.

3.1 Solving Weighted Maximal Matching Problems

The algorithm makes use of a subroutine for solving the weighted maximal matching problem in bipartite graphs, where weights are real values. The matching instances arising in the course of the algorithm may be complete bipartite graphs with a quadratic number of edges, see the counterexample discussed in Section 3.2. Hence, the running times given in the following refer to bipartite graphs with n vertices and $\Theta(n^2)$ edges in order to improve readability. The authors propose to use the algorithm by Hopcroft and Karp [9] to solve an instance of the problem in time $\mathcal{O}(n^{2.5})$. Since this algorithm computes a matching of maximal cardinality, but is not designed to take weights into account, it cannot be applied to the instances that occur.

The best known approaches for the weighted problem allow to solve instances with n vertices and $\Theta(n^2)$ edges in time $\mathcal{O}(n^3)$, e.g., the established Hungarian method [3]. When we assume weights to be integers within the range of $[0..N]$, scaling algorithms would become applicable such as [6], which solves the problem



(a) A star graph H (b) Rooted graph H^c (c) Part in $\mathcal{B}_H$

Figure 2: (a) A star graph of order $n+1$, (b) the star graph rooted at the center vertex, and (c) an elementary part $H^c \setminus \{v_2\}$ obtained from H^{v_2}, where the gray vertex with its incident edge is deleted.

in time $\mathcal{O}(n^{2.5} \log N)$. This running time is still worse than the time bound for the algorithm by Hopcroft and Karp by a factor depending logarithmically on N. Moreover, it is desirable to allow that the weight of a common subgraph graph is measured by a real number depending on the labels of the vertices and edges it contains, cf. [18, Definition 2]. This leads to real edge weights in the matching instances.

In summary, no better bound than $\mathcal{O}(n^3)$ on the worst-case running time can be assumed for the subproblem of solving weighted maximal matching instances with n vertices.

3.2 The Number of Matching Instances

We consider a particularly simple counterexample to illustrate that the running time required to solve the matching problems cannot be bounded by $\mathcal{O}(n^{2.5})$. We identify the flaw regarding the analysis which led to this incorrect result [18, Proof of Theorem 2]. More precisely, we show that for two graphs G and H of order n the BBP-MCS algorithm performs $\Theta(n)$ calls to the subroutine for weighted maximal matching [18, Algorithm 2, MaxMatch] with instances of size $\Theta(n)$. Since the relationship between the matching instances is not considered in [18], we assume that each instance is solved separately in cubic time, cf. Section 3.1. Therefore, no better bound than $\mathcal{O}(n^4)$ can be given on the total running time.

Let the two graphs G and H both be star graphs of order $n+1$, i.e., trees with all but one vertex of degree one as depicted in Figure 2(a). Since trees are outerplanar, G and H are valid input graphs for BBP-MCS. The algorithm presented in [18] relies on a decomposition of the two input graphs into their parts.[2] $\mathit{Parts}(T^r)$ of a rooted tree T^r is recursively defined as follows [18, Definitions 20, 23, 26].

(i) $T^r \in \mathit{Parts}(T^r)$,

[2]The approach greatly simplifies for trees and we have shortened the required definitions accordingly. Please note that [18, Algorithm 4 and Algorithm 3, lines 11-18] will not be required to solve the problem on trees.

(ii) if $P^p \in \mathit{Parts}(T^r)$ and p is incident to exactly one edge $\{p, v\}$, then the graph $(P \setminus \{p\})^v$ is in $\mathit{Parts}(T^r)$,

(iii) if $P^p \in \mathit{Parts}(T^r)$ and p is incident to the edges $\{p, v_1\}, \ldots, \{p, v_k\}$, $k \geq 2$, then for each edge $\{p, v_i\}$, $1 \leq i \leq k$, the connected component of the graph $P^p \setminus \{\{p, v_j\} \mid j \neq i\}$ containing p as root is in $\mathit{Parts}(T^r)$.

For the first input graph G an arbitrary root vertex r is selected to define its parts. Let G be the star graph, r its center vertex and let $L(G)$ denote its leaves, then

$$\mathit{Parts}(G^r) = \{G^r\} \cup \{(\{r, v\}, \{\{r, v\}\})^r \mid v \in L(G)\} \cup \{(\{v\}, \emptyset)^v \mid v \in L(G)\}.$$

The parts of the star graph are the graph itself, the subgraphs consisting of the individual edges and the subgraphs consisting of the leaves. For the second input graph H, its parts are defined as $\mathit{Parts}^*(H) = \cup_{s \in V(H)} \mathit{Parts}(H^s)$ [18, Definition 27]. Therefore,

$$\begin{aligned}\mathit{Parts}^*(H) =& \{H^s \mid s \in V(H)\} \cup \{(e, \{e\})^c \mid e \in E(H)\} \cup \\ & \{(\{v\}, \emptyset)^v \mid v \in L(H)\} \cup \underbrace{\{H^c \setminus \{v\} \mid v \in L(H)\}}_{\mathcal{B}_H},\end{aligned}$$

where c is the unique center vertex of H and $\mathcal{B}_H$ the subgraphs rooted at c obtained by deleting a single leaf with its incident edge, cf. Figure 2(c).

In order to solve the problem, a variant of BBP-MCS, which requires to map the root of one part to the root of the other, is solved for specific pairs of parts denoted by $\mathit{Pairs}(G, H)$. If the roots of both parts have multiple children, a matching problem between them must be solved. Such parts are referred to as *compound-root graphs* and the parts associated with the children are *elementary parts*, respectively [18]. Note that this is the case for G^r and all the parts in $\mathcal{B}_H$; according to [18, Definition 28] we have $\{G^r\} \times \mathcal{B}_H \subseteq \mathit{Pairs}(G, H)$. For each pair (G^r, Q), $Q \in \mathcal{B}_H$, a weighted maximal matching instance is constructed, where the vertices correspond to the elementary parts of G^r and Q [18, Algorithm 2, RMCSCOMPOUND]. The edge weights are determined by the solutions for pairs of smaller parts and depend on the possibly real-valued weights of vertex and edge labels of the common subgraph. The number of elementary parts of G^r is n, the number of elementary parts of each Q in $\mathcal{B}_H$ is $n - 1$. Hence, each of these matching instances has $2n - 1$ vertices and $n(n - 1)$ edges and thus requires time $\mathcal{O}(n^3)$. The number of such pairs is $|\{G^r\} \times \mathcal{B}_H| = n$. If each matching instance is solved separately, no better bound than $\mathcal{O}(n^4)$ on the total running time of the algorithm can be given and the analysis of [18, Theorem 2] is too optimistic.

Consequently, there must be an error in its proof: The authors claim that every vertex $g \in V(G)$ and every vertex $h \in V(H)$ has at most $\deg(g)$ (resp. $\deg(h)$) elementary parts involved in a maximal matching. While this statement is correct the subsequent analysis does not take into account that there may be up to $\deg(h)$ matching instances of that size for a vertex $h \in V(H)$.

More precisely, the total time spent in RMCSCompound for solving matching instances is claimed to be bounded by

$$T_{\text{comp}} = \sum_{g \in V(G)} \sum_{h \in V(H)} T_{\text{WMM}}(\deg(g) + \deg(h)), \tag{1}$$

where $T_{\text{WMM}}(k)$ is the running time for solving a weighted maximal matching instance with k vertices [18, p. 361]. Actually the procedure considers all pairs of compound-root graphs, where each pair leads to a matching instance containing one vertex for each of the associated elementary parts. The counter example above shows that for a vertex $h \in V(H)$ there may be $\deg(h)$ compound-root graphs with root h, each with $\deg(h) - 1$ elementary parts. In addition, there is one compound-root graph with root h and $\deg(h)$ elementary parts. Therefore, a correct upper bound is

$$T_{\text{comp}}^{\text{corrected}} = \sum_{g \in V(G)} \sum_{h \in V(H)} (\deg(h) + 1) \cdot T_{\text{WMM}}(\deg(g) + \deg(h)). \tag{2}$$

In the counter example the degree of the center vertex is not bounded, which leads to the additional factor of n appearing in $T_{\text{comp}}^{\text{corrected}}$, but not in T_{comp}.

3.3 Exploiting the Structure of the Matching Instances

The matching instances emerging for the counter example are closely related, since the symmetric difference of the elementary parts of $Q_1 \in \mathcal{B}_H$ and $Q_2 \in \mathcal{B}_H$ with $Q_1 \neq Q_2$ contains exactly two elements. It was recently shown that this fact can be exploited by solving groups of similar matching instances efficiently in one pass [5]. This technique was used to show that the maximum common subtree problem can be solved in time $\mathcal{O}(n^2\Delta)$ for trees of order n with maximum degree Δ, thus leading to $\mathcal{O}(n^3)$ worst case time. The same technique can be used to improve the running time of the BBP-MCS algorithm.

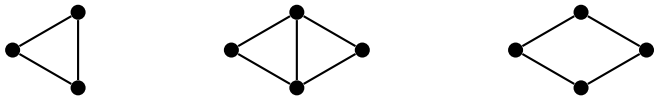
In [5] the proposed maximum common subtree algorithm was compared experimentally to the BBP-MCS algorithm of [18] using the implementation provided by the authors. The running times reported for the BBP-MCS algorithm actually suggest a growth of $\Omega(n^5)$ on star graphs.

4 Violation of the Triangle Inequality

Bunke and Shearer [2] have shown that

$$d(G, H) = 1 - \frac{|\text{Mcs}(G, H)|}{\max\{\, |G|, |H| \,\}}, \tag{3}$$

where $|\text{Mcs}(G, H)|$ is the weight of a maximum common subgraph, is a metric and, in particular, fulfills the triangle inequality. This was originally shown for a definition of the maximum common subgraph problem, which requires common subgraphs to be induced and measures the weight of a graph G by $w(G) =$



(a) G (b) H (c) F

Figure 3: Outerplanar graphs for which Eq. (3) does not satisfy the triangle inequality under BBP-MCS.

$|V(G)|$. Lins et al. [14] proved that Eq. (3) also is a metric for the general (not necessarily induced) subgraph relation, where $w(G) = |V(G)| + |E(G)|$. The article [18] suggests that the weight of a BBP-MCS combined with Eq. (3) is a metric, too. We show that this is not the case.

Consider the example shown in Figure 3 and let the weight of a graph G be defined as $w(G) = |V(G)|+|E(G)|$ following [18, Section 3.2, p. 364]. Employing BBP-MCS, we obtain $|\textsc{Mcs}(G, H)| = 6$, $|\textsc{Mcs}(H, F)| = 8$ and $|\textsc{Mcs}(G, F)| = 1$ and accordingly:

d	G	H	F
G	0	1/3	7/8
H	1/3	0	1/9
F	7/8	1/9	0

The triangle inequality is violated, since $d(G, F) > d(G, H) + d(H, F)$. In general, the connectivity constraints imposed by BBP-MCS make it difficult to derive a metric. For a more detailed discussion of this topic we refer the reader to [11, Section 3.6].

References

[1] T. Akutsu and T. Tamura. A polynomial-time algorithm for computing the maximum common connected edge subgraph of outerplanar graphs of bounded degree. *Algorithms*, 6(1):119–135, 2013. doi:10.3390/a6010119.

[2] H. Bunke and K. Shearer. A graph distance metric based on the maximal common subgraph. *Pattern Recognition Letters*, 19(3-4):255–259, 1998. doi:10.1016/S0167-8655(97)00179-7.

[3] R. E. Burkard, M. Dell'Amico, and S. Martello. *Assignment Problems*. SIAM, 2012. doi:10.1137/1.9781611972238.

[4] M. J. Chung. $O(n^{2.5})$ time algorithms for the subgraph homeomorphism problem on trees. *Journal of Algorithms*, 8(1):106 – 112, 1987. doi:10.1016/0196-6774(87)90030-7.

[5] A. Droschinsky, N. M. Kriege, and P. Mutzel. Faster algorithms for the maximum common subtree isomorphism problem. In P. Faliszewski, A. Muscholl, and R. Niedermeier, editors, *41st International Symposium on Mathematical Foundations of Computer Science (MFCS 2016)*, volume 58 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 33:1–33:14, Dagstuhl, Germany, 2016. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. doi:10.4230/LIPIcs.MFCS.2016.33.

[6] R. Duan and H.-H. Su. A scaling algorithm for maximum weight matching in bipartite graphs. In *Proceedings of the Twenty-third Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA 2012, pages 1413–1424. SIAM, 2012. doi:10.1137/1.9781611973099.111.

[7] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

[8] A. Gupta and N. Nishimura. Sequential and parallel algorithms for embedding problems on classes of partial k-trees. In E. Schmidt and S. Skyum, editors, *Algorithm Theory — SWAT '94*, volume 824 of *Lecture Notes in Computer Science*, pages 172–182. Springer Berlin / Heidelberg, 1994. doi:10.1007/3-540-58218-5_16.

[9] J. E. Hopcroft and R. M. Karp. An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. *SIAM J. Comput.*, 2(4):225–231, 1973. doi:10.1137/0202019.

[10] T. Horváth, J. Ramon, and S. Wrobel. Frequent subgraph mining in outerplanar graphs. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, pages 197–206, New York, NY, USA, 2006. ACM. doi:10.1145/1150402.1150427.

[11] N. M. Kriege. *Comparing Graphs: Algorithms & Applications*. PhD thesis, TU Dortmund, 2015. doi:10.17877/DE290R-16358.

[12] N. M. Kriege, L. Humbeck, and O. Koch. Chemical similarity and substructure searches. In S. Ranganathan, M. Gribskov, K. Nakai, and C. Schönbach, editors, *Encyclopedia of Bioinformatics and Computational Biology*, pages 640 – 649. Academic Press, Oxford, 2019. doi:10.1016/B978-0-12-809633-8.20195-7.

[13] A. Lingas. Subgraph isomorphism for biconnected outerplanar graphs in cubic time. *Theoretical Computer Science*, 63(3):295 – 302, 1989. doi:10.1016/0304-3975(89)90011-X.

[14] L. D. Lins, N. Ferreira, J. Freire, and C. T. Silva. Maximum common subelement metrics and its applications to graphs. *CoRR*, abs/1501.06774, 2015. arXiv:1501.06774.

[15] J. Matoušek and R. Thomas. On the complexity of finding iso- and other morphisms for partial k-trees. *Discrete Mathematics*, 108(1-3):343–364, 1992. doi:10.1016/0012-365X(92)90687-B.

[16] D. W. Matula. Subtree isomorphism in $O(n^{5/2})$. In B. Alspach, P. Hell, and D. Miller, editors, *Algorithmic Aspects of Combinatorics*, volume 2 of *Annals of Discrete Mathematics*, pages 91 – 106. Elsevier, 1978. doi:10.1016/S0167-5060(08)70324-8.

[17] S. W. Reyner. An analysis of a good algorithm for the subtree problem. *SIAM J. Comput.*, 6(4):730–732, 1977. doi:10.1137/0206053.

[18] L. Schietgat, J. Ramon, and M. Bruynooghe. A polynomial-time maximum common subgraph algorithm for outerplanar graphs and its application to chemoinformatics. *Annals of Mathematics and Artificial Intelligence*, 69(4):343–376, 2013. doi:10.1007/s10472-013-9335-0.

[19] R. Shamir and D. Tsur. Faster subtree isomorphism. *Journal of Algorithms*, 33(2):267 – 280, 1999. doi:10.1006/jagm.1999.1044.

[20] M. M. Sysło. The subgraph isomorphism problem for outerplanar graphs. *Theoretical Computer Science*, 17(1):91 – 97, 1982. doi:10.1016/0304-3975(82)90133-5.

[21] R. M. Verma and S. W. Reyner. An analysis of a good algorithm for the subtree problem, corrected. *SIAM J. Comput.*, 18(5):906–908, 1989. doi:10.1137/0218062.